Electronic Version
Stylesheet Version v1.1.1

Description

[Insert title of invention]Process of extracting people's full names and titles from electronically stored text sources

Cross Reference To Related Applications

[0001] This application claims the priority date of the Provisional Patent 60/319,510 filed August 29, 2002.

Background of Invention

[0002] 1. Field of the Invention

[0003] This invention relates to the art of extracting data from electronically stored text sources, more specifically extracting people' s full names and titles.

[0004] 2. Description of Prior Art

[0005]

[0006] Historically, research on companies was done with phone calls, as well as through subscriptions to proprietary databases. Typically these databases contain names and titles of people that work at a company as well as phone numbers. In recent years, email addresses have also been included in these databases. Two examples of database suppliers are Hoovers and Dun & Bradstreet.

[0007] In the mid to late 1990's, a large number of companies started to publish their own company websites on the Internet, accessible via the World Wide Web (WWW). Many of these companies are too small to be included in database directories. Unfortunately, there is not a standard for locating contact information stored within a web site. The only way to find contact information on these web sites is to use a web browser and search through pages. Sometimes a site map is available, but again, there is not a standard.

[0008] It is a common practice for companies to bury contact information several layers deep into their website. For example, a company that sells computers may have a technical support phone number listed, but not on their homepage. Some companies believe that if a person's name or phone number is too accessible, it might be abused. Additionally, a poorly designed web site may also be a challenge to navigate and thus difficult to find information.

[0009] Currently, prior art exists that reads a website and returns a sitemap of the contents of the website. What this accomplishes is essentially providing a sitemap for websites that lack sitemaps. The output from these systems consist of a tree structure breakdown of the web pages on the site. (6,237,006) (*6,144,962).*

[0010] Current art also exists that scans the web pages for email addresses. This is not unique and can be duplicated by any first year computer science student.

SUMMARY OF INVENTION

[0011] The object of the present invention is to provide a method for extracting data from electronically stored text sources, more specifically extracting people's full names and titles.

[0012]]The invention is a process by which peoples names are extracted from electronically stored text. Electronically stored text constitutes any data stream that includes the standard ASCII characters. Examples of data streams are word processor, spreadsheet, or HTML files. The invention can find peoples names stored anywhere within the text of a website or other electronic data repository. A web site can be scanned and names of people listed on the website can be retrieved and stored into a user's database. When a name is identified within a stream of electronic text, additional information such as the person's job title can also be extracted.

[0013] Definitions:

[0014] Whois: A program that will provide the owner's name of any 2nd-level domain name.

[0015] ASCII: American Standard Code for Information Interchange

[0016] WWW: World-Wide Web

[0017] GUI: Graphical User Interface

[0018] HTML: Hypertext Markup Language

[0019] URL: Uniform Resource Locator.

BRIEF DESCRIPTION OF DRAWINGS

[0020] Description of figures

[0021] Figure 1 - Displays a user using the Internet

[0022] Figure 2 - Algorithm extraction states of example name combinations

[0023] Figure 3 - Name extraction algorithm flowchart

[0024] Figure 4 - Name normalization diagram

[0025] Figure 5 - Name probability decrements flowchart

[0026] Figure 6 – Name score Increments list in System

[0027] Figure 7 – Name score Decrements list in System

[0028] Figure 8 – Name score Special cases in System

[0029] Figure 9 – Default name score coefficients in System

[0030] Figure 10 – Formula for final name scoring algorithm

[0031] Figure 11 – Values for X[i], K[i], P[i]

[0032] Figure 12 – Name extraction formula variables

[0033] Figure 13 – Solving the final name scoring formula

[0034] Figure 14 – System Output results

Detailed Description

[0035] The preferred embodiment of the invention is described below.

[0036] The current invention uses Internet communications tool, browser, ISP (Internet Service Providers), embedded website, URL, protocols and languages that are known to one skilled in the art and therefore not disclosed here in detail.

[0037] FIG. 1 illustrates a functional diagram of how a User 10 uses a computer 25 connected to the Internet 500. The computer 25 can be connected directly through a communication means such as a local Internet Service Provider, often referred to as ISPs, or through an on-line service provider like CompuServe, Prodigy, American Online, etc.

[0038] The Users 10 contacts the Internet 500 using an informational processing system capable of running an HTML compliant Web browser. A typical system that is used is a personal computer with an operating system such as Windows 95, 98 or ME or Linux, running a Web browser. The exact hardware configuration of computer used by the User 10 and the brand of operating system is unimportant to understand this present invention.

[0039] Those skilled in the art can conclude that any HTML (Hyper Text Markup Language) compatible Web browser is within the true spirit of this invention and the scope of the claims.

[0040] A computer application that includes the user interface for this invention will be henceforth be referred to as "the system 1." The system 1 focuses on extracting text from HTML pages stored on an internet web site 100. However, the invention is not limited to working with HTML text.

[0041] The System 1 can find peoples names stored anywhere within the text of a website 100. This is a substantial time saver for any User 10 and therefore, it holds significant utility. A web site 100 can be scanned and names of people listed on the website 100 can be retrieved and stored into a user's database. When a name is identified within a stream of electronic text, additional information such as the person's job title can also be extracted.

[0042] The process of extraction relies on multiple component parts that work in conjunction to produce extraction results. Component categories include databases, algorithms, user interface, and output format.

[0043] *Databases elements*

[0044] 1. Names database.

[0045] 2. Additional words databases (top 100 words, top 1000 words)

[0046] 3. Titles database

[0047] 4. Small databases (postal codes, directions, time)

[0048] 5. Famous people database & historic figure database

[0049] *Algorithm s in the system*

[0050] 1. Extraction algorithm

[0051] 2. Substring scoring algorithm

[0052] 3. Final name scoring algorithm

[0053] *User interface elements*

[0054] 1. Substring score - Threshold increments

[0055] 2. Substring score - Decrements

[0056] 3. Substring score - Special cases

[0057] *Output Format*

[0058] 1. The system output

[0059] Before describing the entire invention process, each element must first be defined.

[0060] *Databases elements*Names Database: This is known as the "Names" database. The names database includes over 2 million unique names. A unique name is defined as either a first or a last name. Some entries within the names database are both a first and a last name. Although it is called the names database, it includes more information than just names.

[0061] The names database consists of 7 fields:

[0062] Field 1: NAME: Contains either a first name or a last name.

[0063] Field 2: F: Boolean value that is true if the NAME field is a first name.

[0064] Field 3: L: Boolean value that is true if the NAME field is a last name.

[0065] Field 4: W: W is stored as a 2-byte integer. If W=0, then the NAME field in the same database record is not a word. If W >=1, then the NAME field is a word. Each bit within W denotes a word type (Noun, Verb, etc) that is used by the Substring scoring algorithm. As in the English language, a word can be classified as more than one word type. Example: both a noun and a verb.

[0066] Bit 1: Noun

[0067] Bit 2: Plural

[0068] Bit 3: Noun phrase

[0069] Bit 4: Verb

[0070] Bit 5: Verb Transitive

[0071] Bit 6: Verb Intransitive

[0072] Bit 7: Adjective

[0073] Bit 8: Adverb

[0074] Bit 9: Conjunction

[0075] Bit 10: Preposition

[0076] Bit 11: Interjection

[0077] Bit 12: Pronoun

[0078] Bit 13: Definite Article

[0079] Bit 14: Indefinite Article

[0080] Bit 15: Nominative

[0081] Field 5: A: The value of A determines if the NAME is also an area (city, state, etc.). If A=0, then the NAME field is not an area. If A >=1, then the NAME field is an area. Each bit within A denotes a match for a type of area. For example, a NAME can be both a city and a county.

[0082] Bit 1: NAME is a state or province abbreviation

[0083] Bit 2: NAME is a full state or province name

[0084] Bit 3: NAME is a city

[0085] Bit 4: NAME is a county

[0086] Bit 5: NAME is a country

[0087] Field 6: FF: The frequency that NAME occurs as a first name.

[0088] Field 7: FL: The frequency that NAME occurs as a last name.

[0089] *Additional words databases (top 100 words, top 1000 words):* The additional words databases each have one field. The top 1000 words database contains the 1000 most frequent words found in electronic text. The default form of the top 100 words database is a sub section of the top 1000 words database. Both of these databases are used to ignore frequently used words within electronically stored text. For purposes of speed, both the top 100 and top 1000 databases are embedded into the code of the System 1.

[0090] Titles database: The titles database includes job titles. Examples: President, Chief Financial Officer, Database Administrator.

[0091] Small databases: The small databases are also embedded into the code of the System 1. The small databases include; Postal codes database Contains 548 words listed by the US postal service as being a valid designator of an address (Lane, Road, Way, Annex, etc). Having these available to the extraction algorithm allows the System 1 to ignore names within found addresses. Example: 100 Mike Henry Blvd.

[0092] Directions database: Contains terms that designate direction. (North, South, Up, Down). These also help the algorithm ignore unwanted information.

[0093] Time database: Contains terms that designate time (Today, Daily, Noon)

[0094] *Famous people database & historic figure databases:* These databases are used to identify frequently used names such as "George Bush" to be recognized as text that does not constitute contact information. The names are not ignored as some people are named after famous people. However, it is used to change the statistical significance of the names found within text.

[0095] *Algorithms in the system* Extraction algorithm: The extraction algorithm is the part of the System 1 that scans a stream of electronic text and returns strings that match the criteria of a name. *Figure 3* shows a flowchart illustrating the states of the extraction algorithm. *Figure 4* shows the name normalization process that is sometimes used in conjunction with the extraction algorithm.

[0096] Substring scoring algorithm: The Substring scoring algorithm examines the string retrieved by the extraction algorithm and assigns it a numeric rank. All substrings processed by the Substring scoring algorithm start with the same value. A series of increments and decrements are then applied to the substring. *Figure 5* shows an example of the decrements applied by the Substring scoring algorithm.

[0097] Final name scoring algorithm: Once each substring is scored by the substring scoring algorithm, the values for the name part coefficients are applied to the final scoring algorithm. *Figure 10* shows the formula used by the final name scoring algorithm. *Figure 9* shows the 6 coefficients (PRE, FIRST, MIDDLE, LAST, ANCESTOR, POST). It should be noted that the term "FIRST2" is used interchangeably with the term "MIDDLE.," The "MIDDLE" label is used in the systems 1 user interface and the "FIRST2"label is used by the systems 1 internal processes.

[0098] *User Interface elements* All User 10 interface elements described in this section are intended to be for an administrator level user. An administrator level user is a User 10 who has the rights to install the System 1 on a stand alone computer or computer network. Once the System 1 is installed, user interface elements are not editable. All variables set within the user interface of the System 1 are tied directly to the internal workings of the System 1 algorithms. User editable elements are shown *figures 6,7,8.*

[0099] Increments: The frequency threshold increments are included in a user-editable grid that includes a list of frequency threshold values. Frequencies are stored in the Names database in the field FF and FL. Next to each frequency threshold is an increment value *(figure 6)*. The substring scoring algorithm uses the increment values to increase the score of names found by the extraction algorithm. For example, the first name "John" has a frequency of 2,224,000 in the names database. The number 2,224,000 is larger than the highest frequency threshold (largest increment is 85), so "John" as a first name would get an increment of 85. "John" has a last name frequency of 9000 (greater than 5,000, but less than 10,000). The increment for "John" as a last name would be 45.

[0100] The user-editable grid allows modification of frequency thresholds, and therefore makes the System 1 more flexible. The preferred default values of the grid are shown in *figure 6*.

[0101] Decrements: Decrements are used to lower the ranking of substrings found extracted from text. Using decrements, names that have questionable elements in them are separated from pure names. Decrements are shown in *figure 7*. A pure name is a name in which no substring element is subject to a decrement. Decrements can be applied in the following ways; (1) As individual word within a name such as "Amber"("Amber" is both a word and a name) in the name "Amber Smith;"(2) applied to the entire name such as "George Bush." Each decrement, when true, decreases the substring score by the corresponding value set in the System 1 user interface.

[0102] List of decrements:

[0103] Not caps: A word in an extracted name is not capitalized. Example "john Smith"

[0104] Area: The extracted name is also an area. Example; "Roberta Georgia" can be a woman's name and it is also a city in the state of Georgia.

[0105] Word: The extracted name contains a word.

[0106] Time: The extracted name contains a word in the time database.

[0107] Direction: The extracted name contains a word in the direction database.

[0108] Postal code: The extracted name contains a word in the postal code database.

[0109] State: The extracted name contains the name of a state.

[0110] State abbreviation: The extracted name contains a state abbreviation.

[0111] Famous person: The extracted name is listed in the famous person database.

[0112] Historic figure: The extracted name is listed in the historic figure database.

[0113] Special cases & values: Special case thresholds are used by the extraction algorithm and the substring scoring algorithm. See *figure 8*.

[0114] Name recognition threshold: Minimum value of a final name score required for the System 1 to display an extracted name.

[0115] Threshold area + first: If a first name is an AREA and the frequency of the first name is less than N1, then ignore the name. N1 = value set in user interface.

[0116] Threshold area + last: If a last name is an AREA and the frequency of the last name is less than N2, then ignore the name. N2 = value set in user interface.

[0117] Word + small frequency: If a first or last name is a WORD and the frequency of the name is less than the set value, and then ignore the name.

[0118] Sequential words + top 1000: If 2 sequentially extracted names are both WORDS and one of the 2 words is in the top 1000, then cut off the first word and re-enter the extraction algorithm.

[0119] Top 100: If a name includes a word in the top 100, then cut off the first word and re-enter the extraction algorithm.

[0120] *How all the component parts work together to create the system:*

[0121] *Figure 2* shows combinations of the name of Mr. Michael Joseph Smith-Guterez III PhD as it could appear in electronically stored text. Combinations include names in First Name-Last Name format and Last Name-First Name format. The example name is being used because it includes all possible name part coefficients. "Guterez"is not present in combinations listed in *figure 2* . It is not considered a separate name by the extraction algorithm. It was included in the initial example to show the full extraction scope of the System 1.

[0122] Using *Figure 2*, the extraction algorithm flowchart *(figure 3)* can be traced for any name combination. Use the "Extraction Algorithm States" column from figure 2 as a guide for algorithm flow.

[0123] The name extraction algorithm has 8 possible states (1-8) and 4 special cases (A-D). Each state represents a currently extracted string that contains a name or part of a name. For example, if the System 1 algorithm is at state # 1 the only possible string that can exist is the PRE part of a name. A PRE name part includes designations such as

Mr., Mrs., and Dr. In each state *(figure 3)* values represented in brackets are optional for that state. Values without brackets are required. For example, in state # 4, PRE is optional and both occurrences of FIRST_I are required. FIRST_I represents either a first name or initial. Example name substrings that can be found at state # 4 are the following:" Michael Joseph","M. Joseph","Michael J.","M. J","Mr. Michael Joseph","Mr. M. Joseph","Mr. Michael J.","Mr. M. J".

[0124] In *figure 2*, the different combinations of the POST name coefficient and ANCESTOR name coefficient are shown under the title "Post/Ancestor Combinations". The POST name coefficient is represented in the extraction algorithm as state # 7. The ANCESTOR name coefficient is represented in the extraction algorithm as state # 8. POST and ANCESTOR states have 3 possible combinations that are always appended to the end of the last name. The 3 combinations are shown in *figure 2* under "Post/Ancestor Combinations." Using *figure 2* as a guide, any combination of the example name can be traced through states in the extraction algorithm *(figure 3)*. For example, the combination, "Mr. Michael J. Smith" can be traced from states 1, to 2, to 4, to 6.

[0125] The flowchart of the extraction algorithm *(figure 3)* has 4 locations where a name substring can exist in LAST-FIRST format (after states 3 & 5). In each of these cases, the name must be normalized into FIRST-LAST format. *Figure 4* outlines the normalization process.

[0126] For future clarification, the term "final name scoring formula" refers to the mathematical formula used by the final name scoring algorithm. The "final name scoring algorithm" refers to the implementation of the "final name scoring formula" within the System 1.

[0127] The final name scoring algorithm enables the System 1 to give a numeric score to each name extracted by the name extraction algorithm. If the score is greater than the name recognition threshold (set in the System 1 user interface), then the name is extracted and output by the System 1. If the final name score does not meet name recognition threshold, the first substring of the extracted name is ignored. The name extraction algorithm is then restarted, starting the process over at the second word in the skipped name. The formula used in the final name scoring algorithm is represented in *figure 10*. The breakdown of each variable from the final name scoring formula is shown in *figure 11*.

[0128] In *figure 10*, variable X[i] contains Boolean values representing the presence or absence of a name part. If the name part is found in the extraction process, then X[i] = 1, otherwise X[i] = 0.

[0129] Variable K[i] contains the coefficient values for the name part. Coefficients values are defined in the System 1 user interface *(figure 9)*.

[0130] Variable P[i] represents the probability value set for each name part. The value for P is determined in the name extraction algorithm *(figure 3)*. P[i] is set by the substring scoring algorithm.

[0131] *Figure 12* shows the example name; "Mr Donato S. Diorio"extracted by the name extraction algorithm and then scored by the final name scoring algorithm. The name is divided into component substrings by name part coefficients. Each substring is represented by a different row. Values are shown for X[i], K[i], and P[i].

[0132] Using the final name scoring formula in *figure 10*, and the values from the example name in *figure 1 2*, the expanded formula would take the form shown in *figure 13*.

[0133] Title extraction:Once a name is extracted and it's score is above the name recognition threshold, a title is then scanned for. Scanning for job titles is accomplished by comparing the text directly before and directly after and an extracted name and comparing it to a database of existing titles. Multiple titles may match substrings in proximity to the extracted name. For example: the title "Vice President of Sales" also contains the substring "Vice President" which is also a title. As a rule, the System 1 chooses the longest matching substring for the extracted title. In this example, the System 1 would choose "Vice President of Sales."

[0134] *The system output*

[0135] Once an extracted name has a score, it is saved by the System 1 and later output when scanning is complete. *Figure 14* shows a table of output results from the System 1. Output results from the System 1 are in HTML format and can be viewed with a web browser. In this example, the System 1 scanned an entire web site of a target company.

[0136] Each row of data includes columns;

[0137] Source: The source of the data. Source tells the User 10 where the name was found. For example, names can be found within whois information gathered from a whois server, or a name could be from scanning a web site

[0138] Name: The extracted name and optional title of a person.

[0139] Context: The context the name was found in. Showing the context is crucial for determining if the extracted name is a person related to the web site. In *figure 1 4*, the context for the extracted name "Peter Weddle"(row #7) shows that he is an author. Context gives the User 10 the information to make a choice as to if the name is significant.

[0140] Location: the location is the web page URL that the name was found in.

[0141] The output is arranged so the User 10 of the System 1 can quickly see people's names and titles that were extracted. Names are highlighted in green text and titles in red text.

[0142] *Advantages*

[0143] The previously described version of the present invention has many advantages. The System is a better method of extracting data from electronically stored text sources, especially from web pages.

[0144] Although the present invention has been described in considerable detail with reference to certain preferred versions thereof, other versions are possible. For example, the functionality and look of the System 1 could be different or new protocols or different data structures can be used or different databases could be used. Therefore, the point and scope of the appended claims should not be limited to the description of the preferred versions contained herein.

[0145]